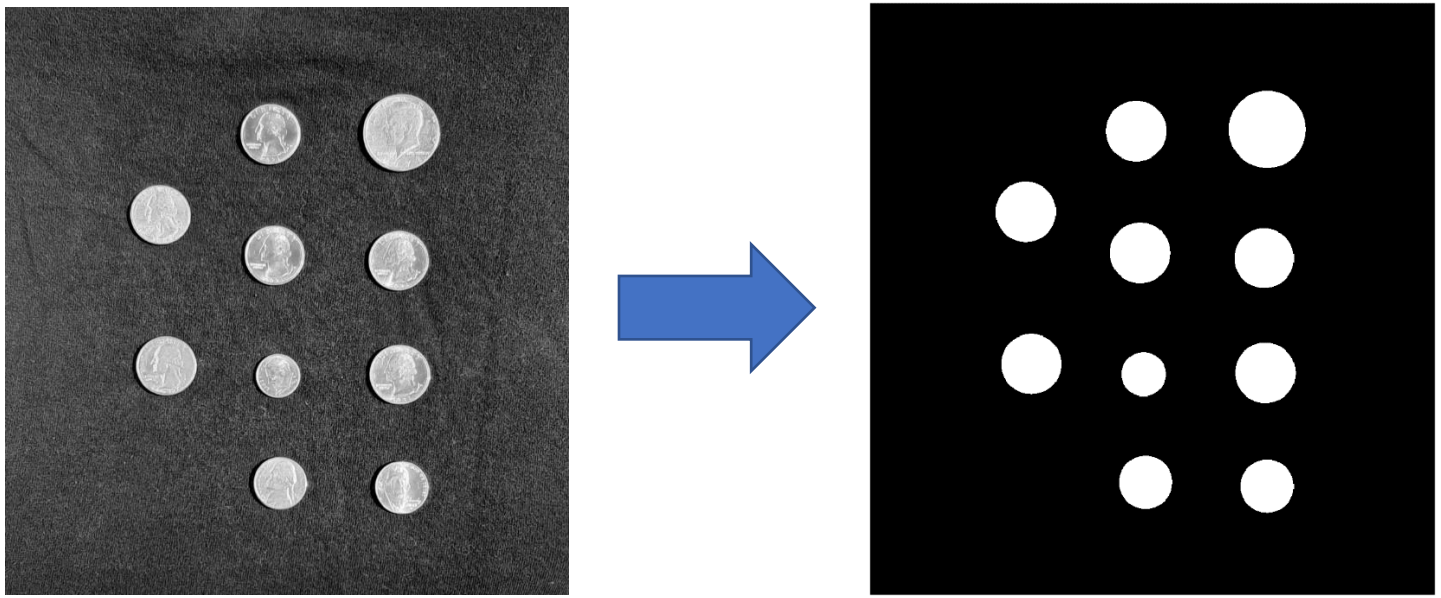
Counting US Coins by Size:

US coin type and value can be determined by their relative size in an image.

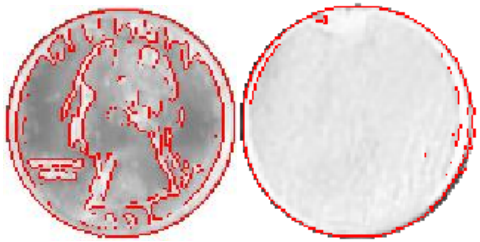
Using this fact, you previously developed code to segment US coins in an image, count each coin type, and determine the total value using the mask region properties.



Determining Valid vs. Blank Coins

Now, what if some of the coins are not valid currency, but rather blanks of the same size?

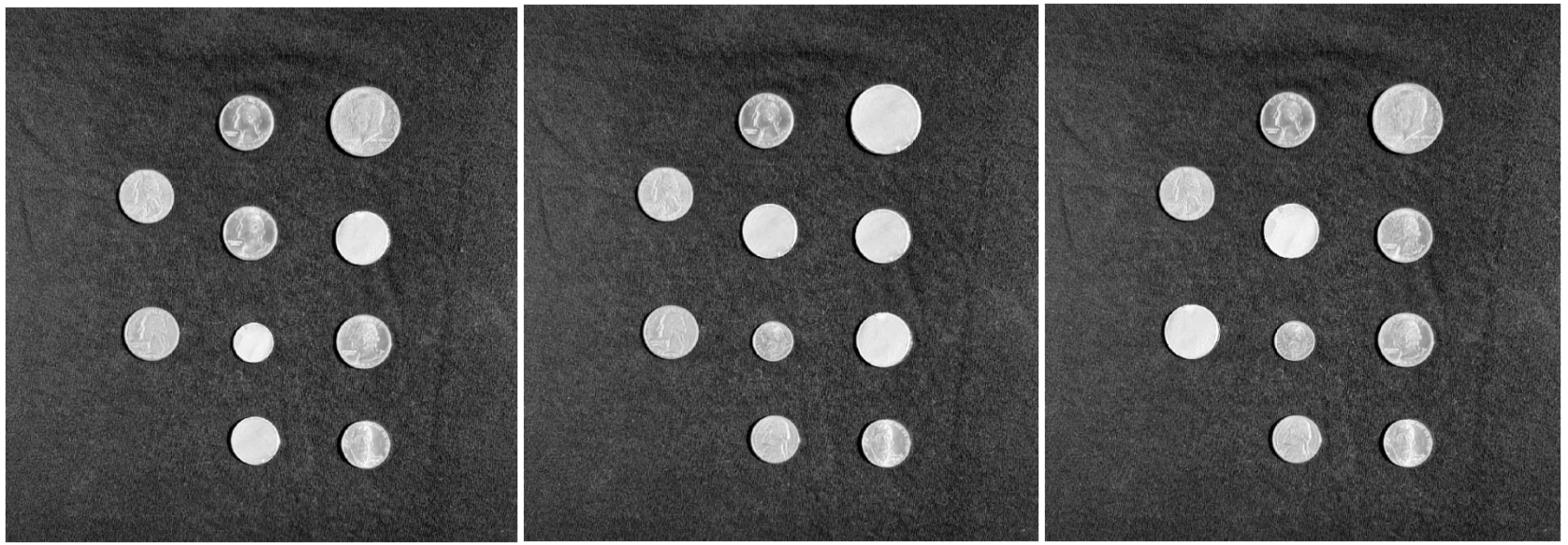
Using the mask region areas and perimeters as before will no longer be sufficient to count the number of each coin. You'll need to differentiate the blank coins from the valid coins. One feature that distinguishes the valid coins from the blanks is that valid coins have more pronounced edges near the center:



In this final project, you'll use the presence or absence of these edges to determine if a coin-sized region in an image is valid currency or a blank. Then you'll count the valid coin types and determine the total value.

Project Tasks

You'll work with three images, each with different coins replaced by blank versions:



There are four main stages in the workflow:

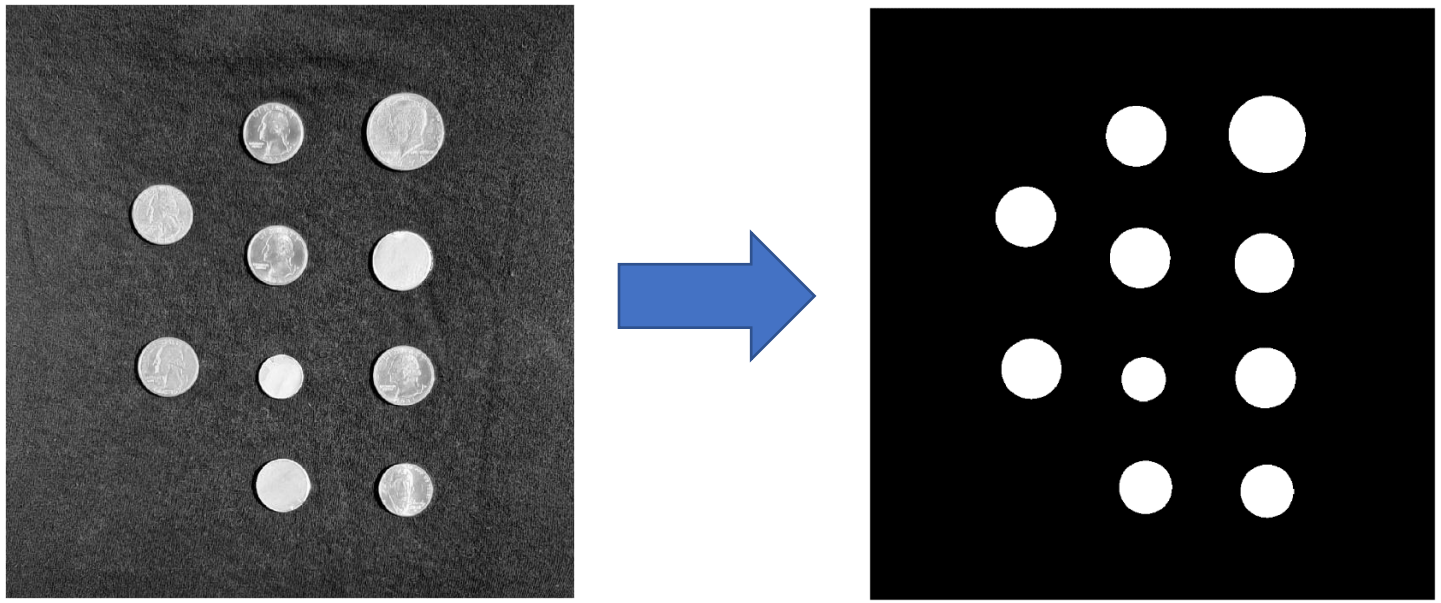
1. Segmenting the foreground (coins and blanks) from the background
2. Detecting and isolating valid coin faces
3. Segmenting only valid coins
4. Counting the number of each coin and calculating the total $ value

You will be assessed in MATLAB Grader at each stage using a random selection of the three test images. Each stage will most likely require you to copy the solution from the previous stage and add code for the new task

We encourage you to develop and test your code in MATLAB before submitting your solutions. The test images are provided with your course files: **testCoinImage1.png**, **testCoinImage2.png**, and **testCoinImage3.png**.

1. Segmenting Foreground

In step one, you need to create a mask that accurately segments the foreground, i.e., both the coins and blanks, from the background. Depending on how you previously segmented the image with all valid coins, the same code may work here, but you'll need to test it.



Using Code Generated from Apps

If you use an app to create the mask and masked image, you can bring your work into MATLAB Grader by generating a function and pasting the entire code at the bottom of the Script box. For example:

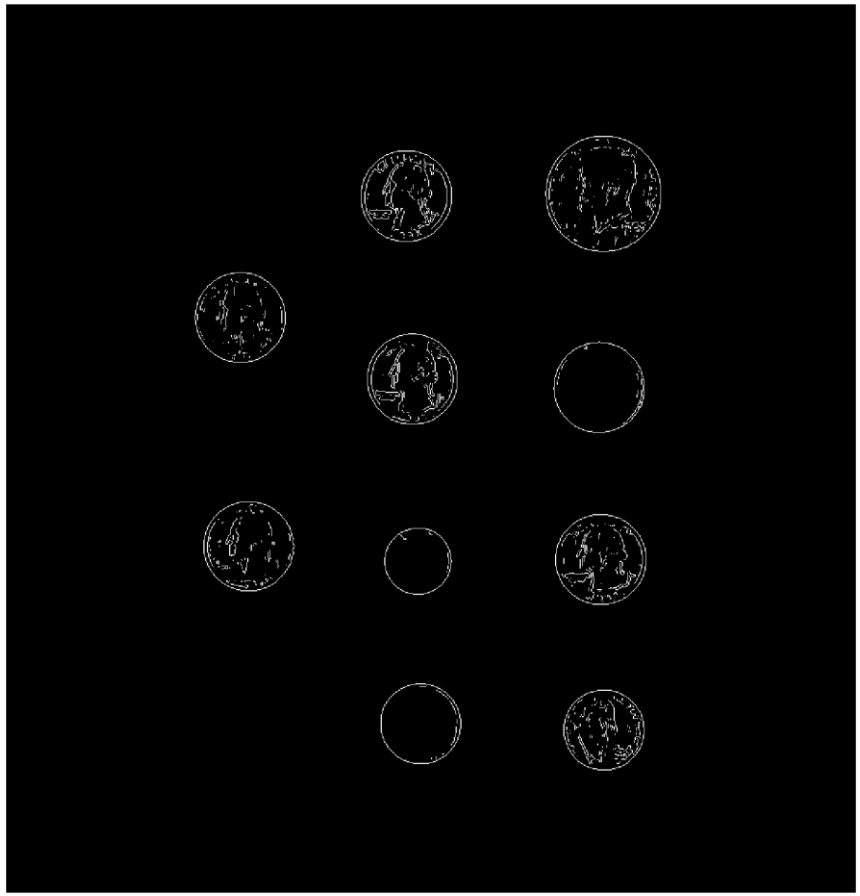
```
x = 1;
y = myFun(x);

function output = myFun(input)
% Auto-generated by an app on 21-Oct-2015
    output = 2*input;
end
```

2. Isolating Coin Face Edges

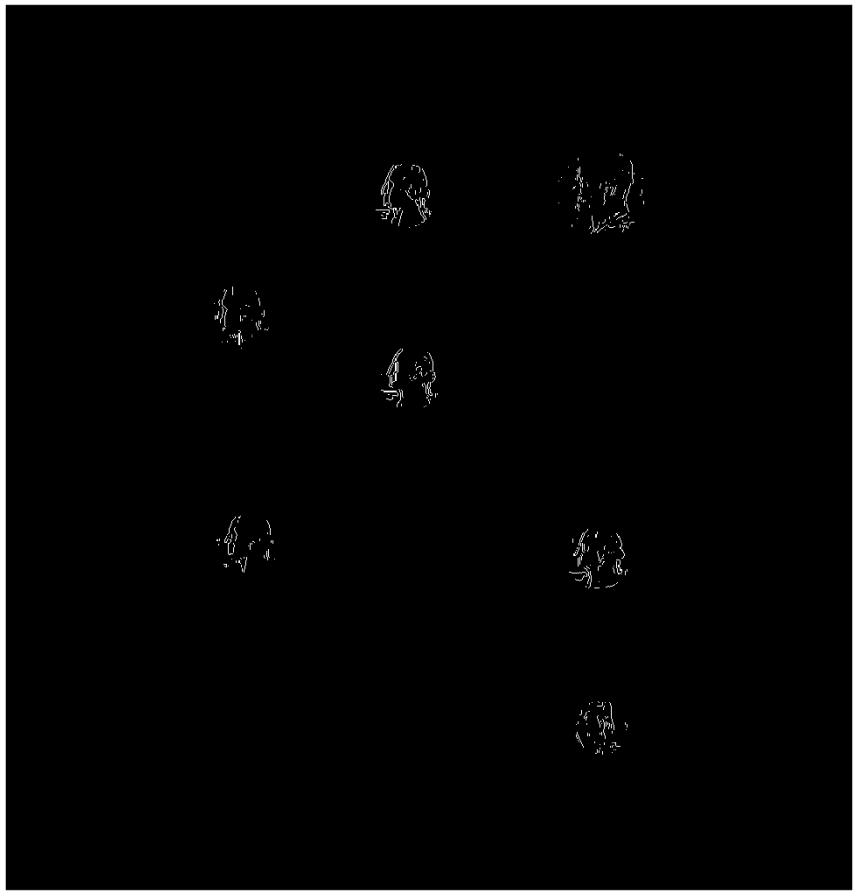
At this stage, you need to create mask with true pixels in the interior of each valid coin, but nowhere else. One workflow to accomplish this is given below:

- Create a binary image with strong edges on the surfaces of the valid coins and few to no edges on the surfaces of the blanks:



NOTE: The background of the original image is not smooth. Background edges could bias an automatically chosen edge threshold. Here, we used the masked foreground image from step one for edge calculations.

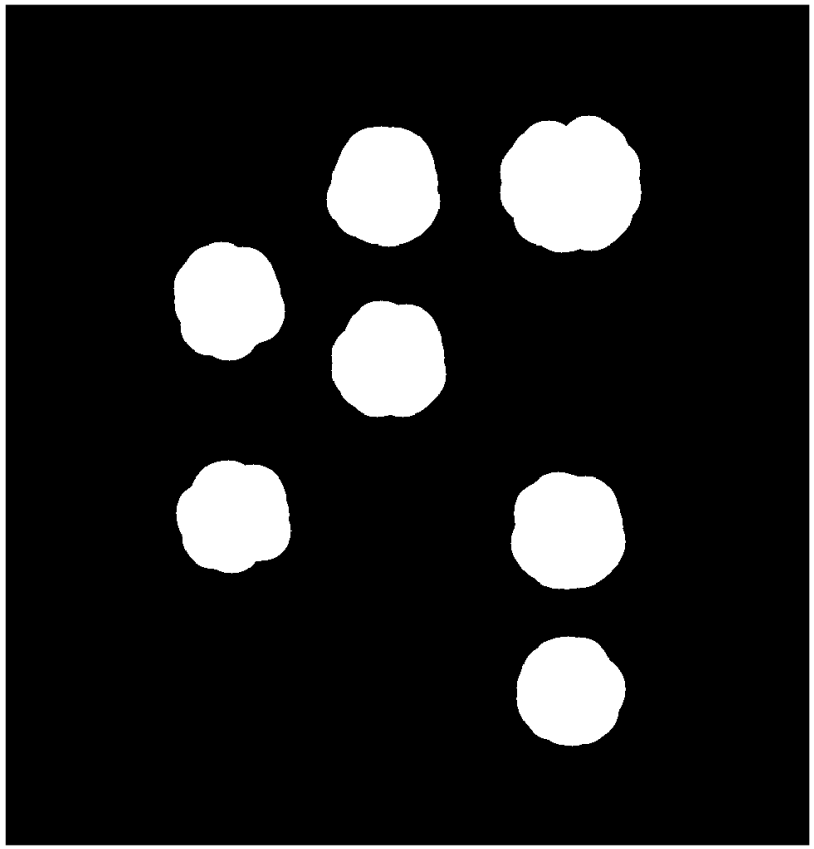
- Eliminate the pixels in the edge mask other than those in the valid coins. Logically combining your edge mask with an eroded version of your foreground segmentation mask should leave you with only the edges closer to the coin centers:



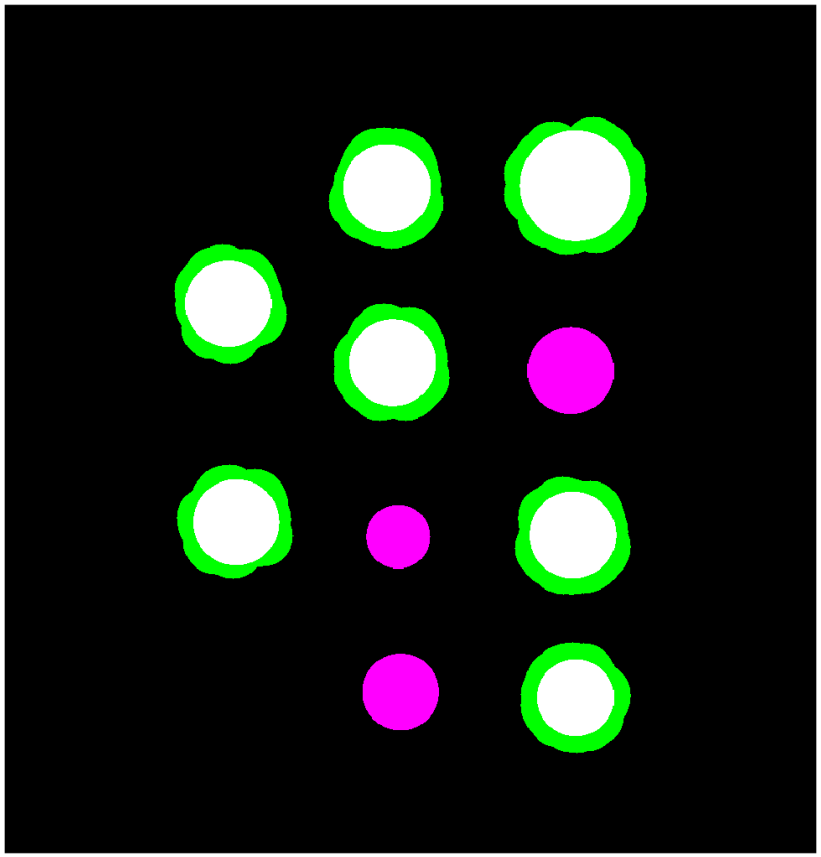
3. Creating a Valid Coin Mask

In this step, you need to create a mask that segments the valid coins. If you did step two correctly, then the valid coin regions are those that overlap the true pixels in your coin face edges mask. One approach to this task is given below:

- Expand the true pixel regions in the coin face edge mask to be greater than or equal to the size of the corresponding foreground regions:



- Combine the result with the foreground mask to extract the valid coin regions (white) and eliminate the blanks (magenta):



4. Counting Valid Coins

Finally, you need to:

- use the sizes of the valid coin regions to determine the type of coin each corresponds to
- count how many coins of each type are present in the image
- calculate the total value of coins in the image

You should be able to use your previous coin counting code for this final task.